

开发框架-自动更新组件 使用说明书

V1.0

目 录

1.	引言	2
1.1.	背景.....	2
1.2.	编写目的.....	2
1.3.	参考资料.....	2
1.4.	术语与缩写	3
2.	自动更新组件.....	3
2.1.	系统上下文关系	3
2.2.	使用效果介绍	3
2.3.	参数配置.....	4
2.3.1.	客户端配置.....	4
2.3.2.	服务端配置.....	5
2.3.3.	自动升级压缩包生成	5
2.4.	客户端代码调用	6

1. 引言

1.1. 背景

在网络化的环境中，特别是基于互联网发布的 Winform 程序，程序的自动更新功能是比较重要的操作，这样可以避免挨个给使用者打电话、发信息通知或者发送软件等，要求其对应应用程序进行升级。实现程序的自动更新，在后期发布维护过程中，一个可以快速相应修正相关的 Bug 或者增加功能，二个可以避免给使用客户隔三差五的发送修改好的程序，省心省力。

基于这个思想，我在 Winform 开发框架中再引入一个自动升级更新的通用模块，这个自动升级的通用模块除了具备一般的功能外，可以通过配置程序标题、升级路径方式等方式，实现通用化的升级操作，其他需要升级的程序，拿过来就可以使用，经过项目的整合及优化，已经具备较好的应用前景了。

本通用自动更新模块具有下面几个特点：

- 1) 程序标题可配置，更新路径可配置，适用于各种需要更新的项目整合。
- 2) 支持文件复制或者对压缩包进行解压，适合更多琐碎程序集的整体升级。
- 3) 支持文件更新后自动启动，或者带参数的启动方式。

1.2. 编写目的

本文档主要介绍《Winform 开发框架》或者《WCF 开发框架》中应用到的自动更新组件的操作使用。

1.3. 参考资料

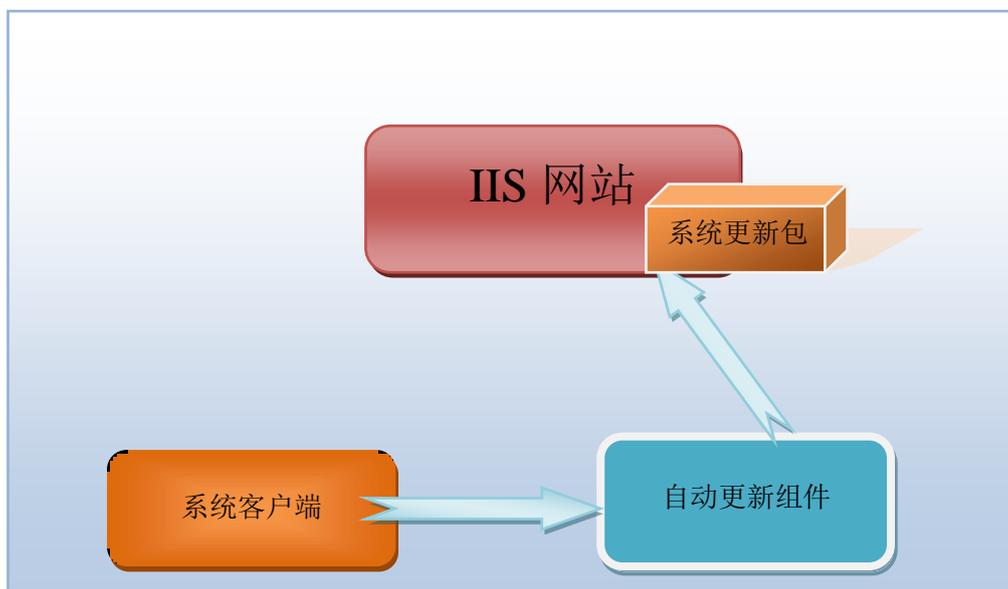
序号	名称	版本/日期	来源
1	《Winform 开发框架-架构设计说明书.doc》		内部
2			内部
3			内部
4			内部

1.4. 术语与缩写

- 1 在本文件中出现的“系统”一词，除非特别说明，均适用于《WCF开发框架》、《Winform开发框架》。
- 2 在本文安装.NET框架中，除非特别说明，均指.NET 4.0框架。

2. 自动更新组件

2.1. 系统上下文关系



自动更新组件是一个独立的可执行文件，在系统客户端中集成引用，可以调用自动更新组件判断是否有版本更新，如果有，则可以由系统客户端主动发起下载更新，自动更新组件根据地址到 IIS 网站中请求下载系统更新包，然后下载到本地，并进行解压缩，完成后自动重启系统客户。

2.2. 使用效果介绍

- 1) 下面是我的 Winform 开发框架中集成通用自动更新模块的截图，如下所示。



- 2) 更新过程中会先关闭主程序，把更新的 Zip 文件下载后进行自动解压，然后启动主程序。

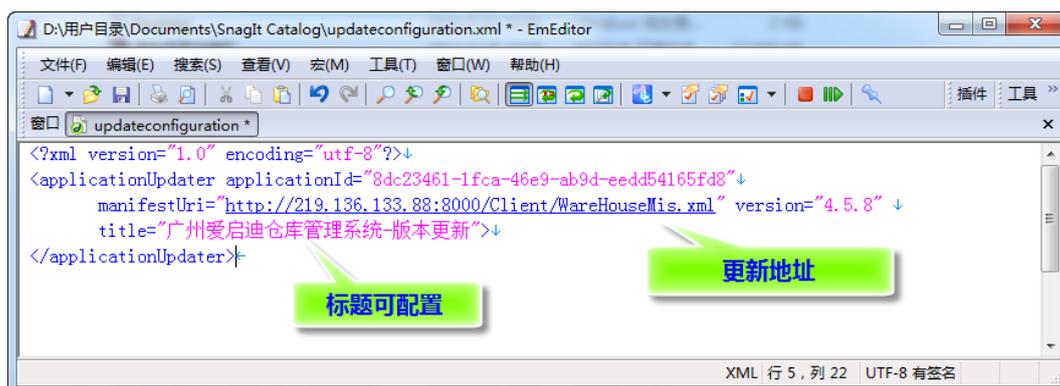


2.3. 参数配置

要实现自动更新模块的正常更新操作，需要配置好客户端配置和服务端配置，这样才能正常进行使用，参数配置也很简单。

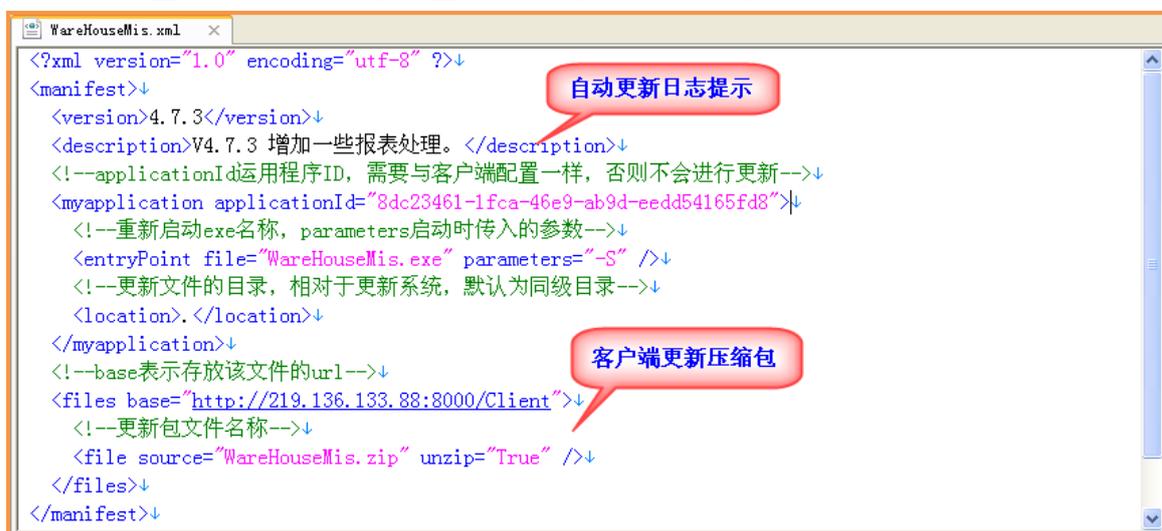
2.3.1. 客户端配置

自动更新“客户端配置”一般需要设置一些参数，如程序标题、更新地址路径、版本号等，自动更新组件的“客户端配置”XML 文件如下所示。



2.3.2. 服务端配置

除了配置客户端外，还需要配置服务端的参数，服务端参数如下所示。



2.3.3. 自动升级压缩包生成

原则上，放到服务器端的程序更新文件，是把版本升级过程中涉及到的变化文件或者程序集压缩成一个 **Zip** 格式的压缩包（如 WareHouseMis.zip），尽可能减少压缩包的大小，因为更新下载太大的文件，耗时会比较长。

注意：另外，由于自动升级模块本身使用了解压缩的类库 **ICSharpCode.SharpZipLib.dll**，因此自动升级压缩包不可包含下面三个文件：**ICSharpCode.SharpZipLib.dll**、**Updater.exe**、**Updater.Core.dll** 三个文件，否则可能出现解压失败的情况，导致程序包无法有效更新。

压缩包下载到客户端后，如果顺利更新文件，那么会删除下载到本地的压缩包，如果因为文件冲突没有顺利解压，那么解压包依旧留在本地，不会删除。

2.4. 客户端代码调用

自动更新虽然可以独立进行执行并更新，不过一般会在主程序中加入对自动更新的判断（毕竟使用客户大多数都是会运行主程序的），实现自动更新判断及执行，我们可以把它放到一个后台线程中执行判断，这样可以提供用户的体验，不会中断界面操作，我一般倾向于把自动更新放到登录界面的首页上，这样用户每次登录的时候，可选择性进行更新，登录后一般要进行业务操作，如果更新退出可能会导致客户的一些重要数据没有保存而丢失，这样影响不好。

C#代码（具体可参考系统的代码）

```
/// <summary>
/// 窗体加载实现的操作
/// </summary>
private void Logon_Load(object sender, EventArgs e)
{
    .....

    #region 更新提示/判断是否自动更新
        updateWorker = new BackgroundWorker();
        updateWorker.DoWork += new DoWorkEventHandler(updateWorker_DoWork);
        updateWorker.RunWorkerCompleted += new RunWorkerCompletedEventHandler(updateWorker_RunWorkerCompleted);

        string strUpdate = config.AppConfigGet("AutoUpdate");
        if (!string.IsNullOrEmpty(strUpdate))
        {
            bool autoUpdate = false;
            bool.TryParse(strUpdate, out autoUpdate);
            if (autoUpdate)
            {
                updateWorker.RunWorkerAsync();
            }
        }
    #endregion

    #region 更新提示线程处理
        private void updateWorker_RunWorkerCompleted(object sender, RunWorkerCompletedEventArgs e)
        {
            //MessageUtil.ShowTips("版本更新完成");
        }
    #endregion
}
```

```
private void updateWorker_DoWork(object sender, DoWorkEventArgs e)
{
    try
    {
        UpdateClass update = new UpdateClass();
        bool newVersion = update.HasNewVersion;
        if (newVersion)
        {
            if (MessageUtil.ShowYesNoAndTips(" 有 新 的 版 本 ， 是 否 需 要 更 新") == DialogResult.Yes)
            {
                Process.Start(Path.Combine(Application.StartupPath, "Updater.exe"), "121");
                Application.Exit();
            }
        }
    }
    catch (Exception ex)
    {
        MessageUtil.ShowError(ex.Message);
    }
}
#endregion
```